Abbottabad University of Science & Technology
Department of Computer Science

SOFTWARE DESIGN DESCRIPTION

(SDD DOCUMENT)

for

<PROJECT NAME>

Version 1.0

By

Student Name 1

Student Name 2

Supervisor

Supervisor Name

Bachelor of Science in Computer Science (20xx-20xx)

Table of Contents

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**

Supervised by

<Supervisor's Name>

Signature_____

Introduction

Briefly explain scope of the project covered till now including modules.

Design methodology and software process model

Explain and justify the choice of design methodology being followed. (OOP or procedural). Also explain which process model are you following and why.

System overview

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

Architectural design

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high-level overview of how the system‟s modules collaborate with each other in order to achieve the desired functionality.

Don‟t go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together.

Provide a diagram showing the major subsystems and their connections. **Use a simple Line-Box-Diagram for simpler systems and detailed diagrams (MVC, Client-Server, Layered, Multi-tiered) for complex systems.**

Process flow/Representation

Provide a representation of the flow of **MAJOR processes** of your system in the form of an activity diagram. **DO NOT CREATE ACTIVITY DIAGRAMS FOR LOGIN OR SIGN-UP UNLESS THEY INVOLVE SIGNIFICANT COMPLEXITY**. Include only the major processes.

Design models [along with descriptions]

The applicable models may include:

- Class Diagram
- Sequence Diagram
- State Transition Diagram
- Data Flow Diagram
- Schematic diagram (Hardware projects only)
- Timing diagram (Hardware projects

only) Insert ***applicable*** system models here.

You should be clear about all the concepts used in your diagrams for example for class diagram you should know about aggregation, composition, and inheritance/generalization. Also ensure visibility of all diagrams.

Class diagram and associated models shall only be necessary for object-oriented approach. In case of procedural, create a DFD. Data flow diagram should be extended to 2-3 levels. It should clearly list all processes, their sources/sinks and data stores.

Note: System design should be complete in all aspects. Create any/all diagrams if you need to. A DFD can also be supplemented by a State Transition Diagram depending on the nature of the project.

Hardware projects can include Schematic diagram, System block diagram, timing diagram, Flow charts as replacement of sequence diagram/ Data flow diagram AFTER CONSULTATION WITH THEIR SUPERVISORS. Choice of models must be properly justified.

Data design

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized.

List any databases or data storage items.

Data dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

Algorithm & Implementation

In this section, we take a closer look at what each component does in a more systematic way. Provide a summary of your algorithm for each function listed in procedural description language (PDL) or pseudo code.

If you gave an OO description, summarize each object member function for all the objects listed in PDL or pseudo code. Describe any local data when necessary.

Software requirements traceability matrix

This section should contain a table that summarizes how each software requirement has been met in this document. The tabular format permits one-to-one and one-to-many relationships to be shown.

Table 7 Requirements Traceability Matrix

Req. Number	Ref. Item	Design Component	Component Items
FR01	Class Diagram	ClassName	FunctionName(s)

OR			
FR01	DFD	DiagramNumber/Level	FunctionName(s)

Human interface design

Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

Screen images

Display screenshots showing the interface from the user‟s perspective. These can be hand-drawn, or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

8.2 Screen objects and actions

A discussion of screen objects and actions associated with those objects

Appendix I

- How to design using UML (OOP): For guidance please follow the instructions mentioned in the link: http://agilemodeling.com/artifacts/

- How and when to design ER diagrams: For guidance please follow the instructions mentioned in the link: http://people.inf.elte.hu/nikovits/DB2/Ullman_The_Complete_Book.pdf

- Data flow diagrams: For guidance please follow the instructions mentioned in the link and book:
  - http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm
  - Software Engineering –A Practitioner‟s approach by Roger Pressman
- Architecture diagram: For guidance please follow the instructions mentioned in the link and book:
  - Ian Sommerville – Software Engineering 9th Edition– Chapter 6